```
import tensorflow as tf
import os
import numpy as np


base_dir=r"/content/drive/MyDrive/CSE475/project2/leaf"
```

Pre-processing for VGG-16

```
IMAGE_SIZE=224
BATCH_SIZE=64

train_datagen=tf.keras.preprocessing.image.ImageDataGenerator(
    rescale=1./255,
    shear_range=0.2,
    zoom_range=0.2,
    horizontal_flip=True,
    validation_split=0.1
    )

test_datagen=tf.keras.preprocessing.image.ImageDataGenerator(
     rescale=1./255,
     validation_split=0.1
)



train_datagen=train_datagen.flow_from_directory(
    base_dir,
    target_size=(IMAGE_SIZE,IMAGE_SIZE),
    batch_size=BATCH_SIZE,
    subset='training'
)

test_datagen=test_datagen.flow_from_directory(
    base_dir,
    target_size=(IMAGE_SIZE,IMAGE_SIZE),
    batch_size=BATCH_SIZE,
    subset='validation'
)
```

```
     Found 90 images belonging to 2 classes.
     Found 10 images belonging to 2 classes.
```

```
from tensorflow.keras.layers import Input,Flatten,Dense
from tensorflow.keras.models import Model
from tensorflow.keras.applications.vgg16 import VGG16
from tensorflow.keras.models import Sequential
from glob import glob


IMAGE_SIZE=[224,224]
vgg=VGG16(input_shape=IMAGE_SIZE+[3],weights='imagenet',include_top=False)
vgg.output
```

```
     Downloading data from https://storage.googleapis.com/tensorflow/keras-applications/vgg16/vgg16_weights_tf_dim_ordering_tf_kernels_notop.
     58889256/58889256 [==============================] - 4s 0us/step
     <KerasTensor: shape=(None, 7, 7, 512) dtype=float32 (created by layer 'block5_pool')>
```

```
for layer in vgg.layers:
    layer.trainable=False


folders=glob(r"/content/drive/MyDrive/CSE475/project2/leaf/*")
print(len(folders))
```

```
     2
```

```
x=Flatten()(vgg.output)
prediction=Dense(len(folders),activation='softmax')(x)
```

```
model=Model(inputs=vgg.input,outputs=prediction)
model.summary()
```

```
Model: "model"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 input_1 (InputLayer)        [(None, 224, 224, 3)]     0

 block1_conv1 (Conv2D)       (None, 224, 224, 64)      1792

 block1_conv2 (Conv2D)       (None, 224, 224, 64)      36928

 block1_pool (MaxPooling2D)  (None, 112, 112, 64)      0

 block2_conv1 (Conv2D)       (None, 112, 112, 128)     73856

 block2_conv2 (Conv2D)       (None, 112, 112, 128)     147584

 block2_pool (MaxPooling2D)  (None, 56, 56, 128)       0

 block3_conv1 (Conv2D)       (None, 56, 56, 256)       295168

 block3_conv2 (Conv2D)       (None, 56, 56, 256)       590080

 block3_conv3 (Conv2D)       (None, 56, 56, 256)       590080

 block3_pool (MaxPooling2D)  (None, 28, 28, 256)       0

 block4_conv1 (Conv2D)       (None, 28, 28, 512)       1180160

 block4_conv2 (Conv2D)       (None, 28, 28, 512)       2359808

 block4_conv3 (Conv2D)       (None, 28, 28, 512)       2359808

 block4_pool (MaxPooling2D)  (None, 14, 14, 512)       0

 block5_conv1 (Conv2D)       (None, 14, 14, 512)       2359808

 block5_conv2 (Conv2D)       (None, 14, 14, 512)       2359808

 block5_conv3 (Conv2D)       (None, 14, 14, 512)       2359808

 block5_pool (MaxPooling2D)  (None, 7, 7, 512)         0

 flatten (Flatten)           (None, 25088)             0

 dense (Dense)               (None, 2)                 50178

=================================================================
Total params: 14,764,866
Trainable params: 50,178
Non-trainable params: 14,714,688
_________________________________________________________________
```

```
model.compile(loss='binary_crossentropy', optimizer='adam', metrics=['accuracy'])
```

```
epoch=5

history=model.fit(train_datagen,
                  steps_per_epoch=len(train_datagen),
                  epochs=epoch,
                  validation_data=test_datagen,
                  validation_steps=len(test_datagen)
                 )
```

```
Epoch 1/5
2/2 [==============================] - 1s 361ms/step - loss: 0.1330 - accuracy: 0.9667 - val_loss: 0.0694 - val_accuracy: 1.0000
Epoch 2/5
2/2 [==============================] - 2s 1s/step - loss: 0.0914 - accuracy: 0.9889 - val_loss: 0.0535 - val_accuracy: 1.0000
Epoch 3/5
2/2 [==============================] - 1s 345ms/step - loss: 0.0913 - accuracy: 0.9778 - val_loss: 0.0452 - val_accuracy: 1.0000
Epoch 4/5
2/2 [==============================] - 2s 1s/step - loss: 0.0622 - accuracy: 1.0000 - val_loss: 0.0481 - val_accuracy: 1.0000
Epoch 5/5
2/2 [==============================] - 1s 355ms/step - loss: 0.0640 - accuracy: 0.9889 - val_loss: 0.0507 - val_accuracy: 1.0000
```

```
om keras.preprocessing import image
ort numpy as np

_pred=image.load_img(r"/content/drive/MyDrive/CSE475/project2/leaf/Strawberry_fresh/02caa98d-1c74-43b3-b3ee-e8492998f82a___RS_HL 2090.JPG",ta
```

```
g_pred=image.img_to_array(img_pred)
g_pred=np.expand_dims(img_pred, axis=0)


lt= model.predict(img_pred)

int(rslt)
 rslt[0][0]>rslt[0][1]:
  prediction="Strawberry_fresh"


se:
  prediction="Strawberry_scrotch"
int(prediction)
```


✓ 8s completed at 1:52 AM